CLAIMS

1. A transaction based constraint enforcer for a database system,
for enforcing a set of constraints that governs the integrity of information stored in
the database system, said enforcer being arranged to delay constraint checks until
the end of a transaction by creating a check stack during the course of the
transaction and executing entries on the check stack at the end of the transaction.

2. Constraint enforcer according to claim 1, further comprising
a stack maker module, arranged for creating and updating said check stack, said
stack maker module being operatively connected to a runtime module in the
database system and arranged to receive data from said runtime module.

3. Constraint enforcer according to claim 2, further comprising
an enforcer module, arranged to receive check data from the check stack, to process
the check data received from the check stack, and to provide resulting data to the
runtime module.

4. Constraint enforcer according to claim 3, wherein said constraints are stored
in a conceptual rules module, comprising rules for prescribing permitted states and
transitions that the database can undertake, said conceptual rules module being
operatively connected to said stack maker module, wherein said stack maker module
is arranged to retrieve constraints from said conceptual rules module.

5. Constraint enforcer according to claim 1,
wherein said check stack is stored on persistent or volatile memory.

6. Constraint enforcer according to claim 2,
wherein said stack maker module is further arranged to handle a modify operator as
a delete operator followed by an insert operator.

7. Method for enforcing a set of constraints that governs the integrity of
information stored in a database system, comprising the step of
delaying constraint checks until the end of a transaction by creating a check stack
during the course of the transaction and executing entries on the check stack at the
end of the transaction.

8. Method according to claim 7, further comprising the following steps,
performed by a stack maker module operatively connected to said conceptual rules
module and to a runtime module in said database system:
receiving data from said runtime module, and
creating and updating said check stack.

9.   Method according to claim 8, further comprising the following steps, performed by an enforcer module:

receiving check data from the check stack,

processing the check data received from the check stack, and

5   providing resulting data to the runtime module.

10.   Method according to claim 9, wherein said constraints are stored in a conceptual rules module, comprising rules for prescribing permitted states and transitions that the database can undertake, further comprising the step of retrieving by said stack maker module constraints from said conceptual rules module.

10   11.   Method according to claim 7,

wherein said check stack is stored on persistent or volatile memory.

12.   Method according to claim 8, further comprising the step of handling by said stack maker module a modify operator as a delete operator followed by an insert operator.

15   13.   A database system, comprising

an application program interface, providing a two-way message interface to a user application program,

a runtime module, operatively connected to the application program interface,

a storage engine module, operatively connected to the runtime module,

20   a data storage, operatively connected to the storage engine module, and

a transaction based constraint enforcer, for enforcing a set of constraints that governs the integrity of information stored in the database system, said enforcer being arranged to delay constraint checks until the end of a transaction by creating a check stack during the course of the transaction and executing entries on the check

25   stack at the end of the transaction.

14.   System according to claim 13, wherein said constraint enforcer further comprises a stack maker module, arranged for creating and updating said check stack, said stack maker module being operatively connected to a runtime module in the database system and arranged to receive data from said runtime module.

30   15.   System according to claim 14, wherein said constraint enforcer further comprises an enforcer module, arranged to receive check data from the check stack, to process the check data received from the check stack, and to provide resulting data to the runtime module.

16.   System according to claim 15,

35   wherein said constraints are stored in a conceptual rules module, comprising rules for prescribing permitted states and transitions that the database can undertake, said conceptual rules module being operatively connected to said stack maker module,

said stack maker module being arranged to retrieve constraints from said conceptual rules module.

17.   System according to claim 13,
wherein said check stack is stored on persistent or volatile memory.

5   18.   System according to claim 14, wherein said stack maker module is further arranged to handle a modify operator as a delete operator followed by an insert operator.

10